CLAIMS

What is claimed is:

1. A method for mapping a function to logic in a programmable logic device having at least one look up table (LUT) and at least one dedicated logic element, the method comprising:

factoring the function to derive a factored form of the function; and

implementing the factored form of the function using the dedicated logic element.

2. The method of Claim 1, wherein implementing the factored form of the function using the dedicated logic element comprises:

implementing a first portion of the factored form of the function using the LUT; and

implementing a second portion of the factored form of the function using the dedicated logic element.

3. The method of Claim 1, wherein factoring the function to derive a factored form of the function comprises forming a plurality of factored cube sets for the function, wherein each factored cube set comprises a shared set and an unshared set.

4. The method of Claim 3, wherein the function comprises a plurality of p-terms, and forming a plurality of factored cube sets comprises:

forming a first factored cube set to accommodate a first p-term; and

forming a second factored cube set to accommodate a second p-term when the first factored cube set cannot accommodate the second p-term.

5. The method of Claim 4, further comprising:

forming a third factored cube set to accommodate a third p-term when the first factored cube set and the second factored cube set cannot accommodate the third p-term.

6. The method of Claim 3, wherein implementing the factored form of the function using a dedicated logic element, comprises:

using a first LUT to perform a first function on a first portion of the shared set of a first factored cube set;

using a second LUT to perform a second function on a portion of the unshared set of the first factored cube set; and

forming a first LUT chain by coupling the first LUT and the second LUT using a first dedicated logic element of the programmable logic device.

7. The method of Claim 6, further comprising:

using a third LUT to perform a third function on a second portion of the shared set of the first factored cube set; and

expanding the first LUT chain by coupling the third LUT to the first LUT chain using a second dedicated logic element of the programmable logic device.

8. The method of Claim 6, further comprising:

using a third LUT to perform a third function on a second portion of the unshared set of the first factored cube set; and

expanding the first LUT chain by coupling the third LUT to the first LUT chain using a second dedicated logic element of the programmable logic device.

9. The method of Claim 6, wherein the first dedicated logic element includes a carry element.

10. The method of Claim 7, further comprising sorting variables of the shared set based on arrival time.

11. The method of Claim 7, further comprising calculating a maximum variable arrival time for each LUT.

12. The method of Claim 11, wherein the first LUT chain is arranged based on the maximum variable arrival time of each LUT.

13. The method of Claim 6, further comprising:

forming a second LUT chain for a second factored cube set; and

coupling the second LUT chain to the first LUT chain with a second dedicated logic element of the programmable logic device.

14. The method of Claim 13, wherein the second dedicated logic element is a cascade element that implements the OR function.

15. The method of Claim 13, further comprising:

calculating an associated delay for each LUT chain; and

arranging the LUT chains based on the associated delays.

16. The method of Claim 3, wherein implementing the factored form of the function using a dedicated logic element, comprises:

using a first combination of LUTs to implement a first factored cube set;

using a second combination of LUTs to implement a second factored cube set; and

forming a first LUT chain by coupling the first combination of LUTs and the second combination of LUTs

using a first dedicated logic element of the programmable logic device.

17. The method of Claim 16, wherein the first dedicated logic element is a carry element.

18. The method of Claim 17, wherein the carry element is a multiplexer configured to implement a logical OR function.

19 The method of Claim 17, wherein the carry element is a multiplexer configured to implement a logical AND function.

20. The method of Claim 16, further comprising:

using a third combination of LUTs to implement a third factored cube set; and

expanding the first LUT chain by coupling the third combination of LUTs to the first LUT chain using a second dedicated logic element of the programmable logic device.

21. The method of Claim 16, further comprising calculating a LUT combination delay for each combination of LUTs.

22. A programmable logic device configured to implement a function in factored form having a plurality of factored cube sets, wherein each factored cube set comprises a shared set and an unshared set, the programmable logic device comprising:

a first LUT configured to implement a first portion of the shared set of a first factored cube set;

a second LUT coupled to implement a portion of the unshared set of the first factored cube set; and

a first dedicated logic element coupling the first LUT and the second LUT to form a first LUT chain.

23. The programmable logic device of Claim 22, further comprising:

a third LUT configured to implement a second portion of the shared set of the first factored cube set; and

a second dedicated logic element coupling the third LUT to the first LUT chain.

24. A programmable logic device, configured to implement a function in factored form having a plurality of factored cube sets, wherein each factored cube set comprises a shared set and an unshared set, the programmable logic device comprising:

a first combination of LUTs configured to implement a first factored cube set;

a second combination of LUTs configured to implement a second factored cube set; and

a first dedicated logic element coupling the first LUT and the second LUT to form a first LUT chain.

25. A method of mapping a softPAL into a PLD comprising:

expressing the softPAL as a sum-of-products (SOP) function;

factoring the SOP function into at least one chain of factored cube sets (FCSs) having the form

F(i) = <shared set> * <unshared set>,

where the shared set is a logical AND function of any number of variables, and the unshared set is a logical OR function of K or fewer logically ANDed variables, where K is a number of inputs to a lookup table (LUT) in the PLD; and

implementing the at least one chain of FCSs with a combination of LUTs and dedicated resources.

26. The method of Claim 25 further comprising:

adding product terms to the unshared set to be implemented by a LUT until K is reached, then forming another FCS if additional product terms remain.

27. The method of Claim 25 wherein the combination of LUTs and dedicated resources is selected from several possible combinations to minimize delay of the overall function F.

28. The method of Claim 25 wherein three schemes are considered for the step of implementing the at least one chain of FCSs with a combination of LUTs and dedicated resources.

29. The method of Claim 25 wherein the step of implementing implements at least one portion of the shared set in a carry chain of the PLD.

30. The method of Claim 25 wherein the step of implementing implements at least one portion of the unshared set using cascade OR gates of the PLD.

31. The method of Claim 25 wherein the step of implementing implements at least a portion of the unshared set in a carry chain of the PLD.

32. The method of Claim 25 wherein the step of implementing comprises:

calculating maximum arrival delay of variables in the unshared set;

forming a LUT implementation from the variables in the unshared set; and

combining the LUTs implementing the shared and the unshared sets

33. The method of Claim 25 wherein the step of implementing comprises:

generating a sorted shared variable list by arranging the shared variables in order of arrival times of the shared variables; and

forming a sorted LUT chain by grouping the sorted shared variables into LUTs with earliest arriving variables closest to a beginning of the LUT chain.

34. The method of Claim 25 wherein the step of implementing comprises:

for each FCS,

implementing an FCS chain within a single CLB; and

calculating delay of the FCS chain;

arranging the FCS chains in increasing order of delay of the FCS chain; and

connecting the FCS chains using cascade elements implementing a logical OR function.

35. The method of Claim 34 wherein the step of implementing further comprises:

sorting FCSs in each FCS chain in increasing order of delay of the FCSs in the FCS chain.

36. A method for mapping a function to logic elements in a programmable logic device comprising:

factoring the function to derive a factored form of the function;

determining arrival time for input signals to each factor in the factored form; and

implementing the factored form of the function such that each factor is implemented by one of the logic elements, and factors having signals with the latest arrival times are implemented by logic elements closest to an output location of the function.

37. The method of Claim 36, wherein a determination of which logic elements are closest to an output location is determined by using a cost table specifying delay for each softPAL that implements a factor.

38. The method of Claim 37, wherein the cost table includes softPALs listed in order of increasing delay.

39. The method of Claim 37, wherein each softPAL that implements a factor is selected from a plurality of possible softPALs that can implement the factor.

40. The method of Claim 37, wherein each softPAL is selected from the plurality of possible softPALs based on minimizing delay.

41. A post-mapping optimizing method comprising:

(a) forming a mapped design mapped to a combination of LUTs and original sized softPALs;

(b) calculating original critical path delay in the mapped design;

(c) for each critical path in the mapped design:

re-mapping the critical path node to a larger sized softPAL;

if delay is reduced, accepting the large sized softPAL;

if delay is not reduced, retaining the LUT or original sized softPAL;

(d) re-calculating a new critical path delay for a new mapped design;

(e) if the new critical path delay for the mapped design is less than the original critical path delay, accepting the new mapped design; and

(f) repeating steps (a) through (e) until the new critical path delay for the mapped design is not less than the original critical path delay.

42. A p-term estimation method for a function comprising:

decomposing the function into a function tree of AND nodes, OR nodes, and NOT nodes where each of the nodes has no more than two inputs;

associate all inputs to a node with an array, where the size of the array is a number of p-terms corresponding to an SOP function performed at an input to the node, and a value of each element of the array is a number of variables input to each p-term of the SOP function; and

computing the array associated with an output of the node based on a function performed by the node.

43. The p-term estimation method of Claim 42, wherein if a node performs the AND function, the array is determined using the pseudocode:

```
for (p = 1 to n) {
     for (q = 1 to m) {
          A_NA[(p-1)*m +q] = L[p] + R[q];
     }
}
```

where p and q are integer indices, n represents the number of nodes, and m represents the number of p-terms.

44. The p-term estimation method of Claim 42, wherein if a node performs the OR function, the array is determined using the pseudocode:

```
for (p = 1 to n) {
     A_NO[p] = L[p];
}
for (q = 1 to m) {
     A_NO[n+q] = R[q];
}
```

where p and q are integer indices, n represents the number of nodes, and m represents the number of p-terms.

45. The p-term estimation method of Claim 42, wherein if a node performs the NOT function, the array is determined using the pseudocode:

```
size = 1;
for (p = 1 to n) {
     size = size * L[p];
```

}

where p is an integer index and n represents the number of nodes.

46. A method for determining a delay of a node for implementing a softPAL function using a combination of LUTs and dedicated logic elements, the method comprising:

forming a first set of one or more FCS chains representing the equation;

implementing the first set of FCS chains with a first implementation scheme;

assigning the node a delay of the first implementation scheme;

forming a second set of one or more FCS chains representing the equation;

implementing the second set of FCS chains with a second implementation scheme; and

updating the delay of the node when a delay of the second implementation scheme is less than the delay of the node.

47. A method for decreasing delay of a node in a critical path during a mapping of a function, comprising:

obtaining a first delay for the node based on a first element in a cost table;

assigning the first delay to a delay of the node;

obtaining a second delay for the node based on a second element in the cost table; and

assigning the second delay to the delay of the node when the second delay is smaller than the first delay.

48. A method of mapping a function to a combination of one or more LUTs and softPALs, the method comprising:

forming factored cube sets (FCSs) from the function;

configuring a set of one or more vertical elements to perform a logical function;

mapping the FCSs to a series of one or more LUTs performing logical functions; and

coupling the one or more LUTs with the one or more vertical elements.

49. A method for estimating a number of product terms in a function tree for a node, comprising:

receiving a two-bounded function tree;

determining an array for each child of the node; and

estimating the number of product terms for the node from the arrays of each child of the node based on a type of the node.